Supervised Aggregative Feature Extraction for Big Data Time Series Regression

Susto, G. A., Schirru, A., Pampuri, S., & McLoone, S. (2016). Supervised Aggregative Feature Extraction for Big Data Time Series Regression. *IEEE Transactions on Industrial Informatics*, *12*(3), 1243-1252. https://doi.org/10.1109/TII.2015.2496231

Published in:
IEEE Transactions on Industrial Informatics

Document Version:
Peer reviewed version

Supervised Aggregative Feature Extraction for Big Data Time Series Regression

Gian Antonio Susto, Andrea Schirru, Simone Pampuri, and Seán McLoone *Senior Member, IEEE*

***Abstract*—In many applications, and especially those where batch processes are involved, a target scalar output of interest is often dependent on one or more time series of data. With the exponential growth in data logging in modern industries such time series are increasingly available for statistical modeling in soft sensing applications. In order to exploit time series data for predictive modelling, it is necessary to summarise the information they contain as a set of features to use as model regressors. Typically this is done in an unsupervised fashion using simple techniques such as computing statistical moments, principal components or wavelet decompositions, often leading to significant information loss and hence suboptimal predictive models. In this paper, a functional learning paradigm is exploited in a supervised fashion to derive continuous, smooth estimates of time series data (yielding aggregated local information), while simultaneously estimating a continuous shape function yielding optimal predictions. The proposed Supervised Aggregative Feature Extraction (SAFE) methodology can be extended to support nonlinear predictive models by embedding the functional learning framework in a Reproducing Kernel Hilbert Spaces setting. SAFE has a number of attractive features including closed form solution and the ability to explicitly incorporate first and second order derivative information. Using simulation studies and a practical semiconductor manufacturing case study we highlight the strengths of the new methodology with respect to standard unsupervised feature extraction approaches.**

***Index Terms*—Big Data, Feature Extraction, Machine Learning, Regularization Methods, Semiconductor Manufacturing, Statistical Modeling, Soft Sensor, Time Series Learning.**

I. Introduction

MACHINE learning methodologies are applied in many industrial areas to create models of observable phenomena from representative datasets [27]. Thanks to the increasing availability of data from on-line sensing of processes in modern industries [8], they are assuming a fundamental role in decreasing measurement costs and enhancing process quality. A typical example is provided by *Soft Sensing* technologies [7], [25], [29] that have proliferated for example in biotechnology [19] and manufacturing [27] under a range of different names, including virtual sensing, virtual metrology, statistical sensing and inferential estimation. Soft sensors are statistical models that provide an estimate of quantities (outputs y) that may be unmeasurable or costly/time-consuming to measure based on more accessible 'cheap to measure' variables (inputs $\mathcal{X}$).

In industrial modeling one of the challenges associated with 'big data' is how to condense the information contained therein into a form that is suitable for modeling without incurring significant information loss [6], [16]. In this paper we consider a specific modelling scenario frequently encountered in industrial environments, especially those involving batch production such as chemical [9], [15] and semiconductor manufacturing [27], namely, where the input information for the model is conveyed in the form of time series. The presence of time series input data can increase modelling computational costs by several orders of magnitude due to the explosion in input dimensionality (potentially hundreds or thousands of samples instead of a single value) or it may not even be possible to directly generate the design matrix required for modelling.

In mathematical terms the scenario in question is to identify a model f of the phenomenon under consideration by exploiting a *training dataset* $\mathcal{S}$ of n observations of the phenomenon, where $\mathcal{S}$ is defined as

$$\mathcal{S} = \{\mathcal{X}_i,\ y_i \in \mathbb{R}\}_{i=1}^{n}, \tag{1}$$

with $\mathcal{X}_i$, the i-th observation, consisting of a set of p time series defined as

$$\mathcal{X}_i = [x_i^{(1)}(t)\ \ldots\ x_i^{(j)}(t)\ \ldots\ x_i^{(p)}(t)],\ t \in [0,1], \forall\ j$$

and y_i is a scalar target value. The predictor function f is chosen to be optimal in the sense that, given a set of independent observations of the phenomenon (test set) $\mathcal{S}^* = \{\mathcal{X}_i^*,\ y_i^*\}_{i=1}^{n^*}$, the loss function

$$\mathcal{L} = \sum_{i=1}^{n^*} d\left[f\left(\mathcal{X}_i^*\right), y_i^*\right],$$

where d is a defined distance metric, is minimized.

In practice, the continuous time series $x_i^{(j)}(t)$ are not available, rather they are represented by a set of discrete noise corrupted observations (samples)

$$\left\{t_{i,s}^{(j)},\ z_{i,s}^{(j)}\right\}^{\mathcal{N}_{i,j}}\ ;\ z_{i,s}^{(j)} = x_i^{(j)}(t_{i,s}^{(j)}) + v_{i,s}^{(j)}\ ,$$

G.A. Susto (corresponding author) is with University of Padova, Italy and with Statwolf LTD, Ireland. E-mail: gianantonio.susto@dei.unipd.it.

A. Schirru and S. Pampuri are with University of Pavia, Italy and with Statwolf LTD.

S. McLoone is with Queen's University Belfast, Northern Ireland.

where $t_{i,s}^{(j)}$ and $z_{i,s}^{(j)}$ are the time and value of the s-th sampled point from the j-th time series of the i-th observation and $v_{i,s}^{(j)}$ is the corresponding measurement noise ($v_{i,s}^{(j)} \sim N(0, \rho_j^2)$). In the most general setting the time series may be irregularly sampled and vary in length, that is, $\mathcal{N}_{i,j} \neq \mathcal{N}_{i,m},\ \mathcal{N}_{i,j} \neq \mathcal{N}_{k,j}$ and $t_{i,s}^{(j)} \neq t_{i,s}^{(m)}, t_{i,s}^{(j)} \neq t_{k,s}^{(j)}$.

Regression functional paradigms have previously been considered for time-series data [5], [14], however these tools are generally intended for univariate problems ($p = 1$) or in a few cases [10] low dimensional problems, (i.e. small p), and as such are not suited to industrial modeling problems, where high dimensional and high volume (big data) problems are becoming more and more common place. In order to make identification of a model from the data described in the previous paragraph tractable using machine non-functional learning techniques it is necessary to extract a homogeneous set of features from every observation to use as model inputs. However, it is not possible to know in advance what part of a given time series (if any) has an impact on the target variable. This lack of information must be taken into account when choosing a feature extraction methodology since, in general, the extraction of a set of features from an observation will result in the loss of some information. This is especially true when the format of such information is expected to show inter-example differences. The goal is to build a design matrix $\Phi \in \mathbb{R}^{n \times \overline{p}}$ containing n observations of $\overline{p}$ summary features that can be subsequently used, along with the target variable vector $Y \in \mathbb{R}^n$, to train a predictor using a machine learning algorithm. Hence, the challenge is to aggregate the information contained in each time series so that summary features are produced that are good predictors of the target value.

A standard framework used for time series feature extraction is to partition the input time series into $\mathcal{M}$ intervals $[\tau_1 \ \ldots \ \tau_{\mathcal{M}}]$ and to compute statistical moments up to order k_{max} for each interval; this approach has been used in several soft sensing applications, see for example [9], [18] and [27].

Given p time series, this then allows a design matrix Φ of the form $\Phi = [\Phi_1 \ \ldots \ \Phi_j \ \ldots \Phi_p]$, to be constructed where $\Phi_j \in \mathbb{R}^{n \times k_{\max}\mathcal{M}}$ are sub-matrices populated with the interval-wise statistical moments for each time series. Two common choices within this framework are:

- Setting $\mathcal{M} = 1$ - each time series is represented by a number of global statistical moments (mean, variance, kurtosis, etc.).
- Setting $k_{\max} = 1$ - each time series is represented by a sequence of local averages (downsampling)

Both approaches suffer from major drawbacks. Statistical moments do not take account of the dependency between information and time, while down sampling requires *a priori* selection of the number of segments, $\mathcal{M}$, which is a trade-off between locality (temporal resolution) and stability of information (robustness to noise).

To cope with the aforementioned issues with classical feature extraction approaches we recently proposed a novel methodology [21], referred to as Supervised Aggregative Feature Extraction (SAFE) that exploits a functional learning paradigm in order to derive continuous, smooth estimates of the time series data (yielding aggregate local information), while at the same time estimating a continuous shape function to provide optimal predictions. In this paper, which in an extension of [21], we provide a comprehensive description of SAFE, present new insights on the technique's interpretability qualities, tuning procedure, computation complexity and sensitivity to sampling rate. In addition to more extensive simulation results and comparisons with existing techniques, we also demonstrate the utility of SAFE on a challenging big data industrial case study from the semiconductor manufacturing sector.

The remainder of the paper is organized as follows. Section II introduces some basic machine learning and regularization concepts and notation needed for the derivation of the SAFE methodology which is presented in Section III. Then, in Section IV and V SAFE is compared to a number of classical feature extraction techniques on simulated and real industrial data case studies. Final remarks are provided in Section VI.

II. Preliminaries

Given the design matrix Φ, vector of target outputs Y, and a suitably chosen model structure $f(\Phi, \theta)$, estimation of model parameters θ can be expressed as a regularized optimisation problem

$$\theta_\lambda^* = \arg\min_\theta \ \mathcal{L}_\lambda(\theta), \tag{2}$$

where fitness function $\mathcal{L}_\lambda(\theta)$ is defined as

$$\mathcal{L}_\lambda(\theta) = \mathcal{F}(\theta) + \lambda \mathcal{R}(\theta). \tag{3}$$

Here, $\mathcal{F}$ is a *cost function* which measures the approximation error of $f(\Phi, \theta)$ over the training data $\mathcal{S}$ and $\mathcal{R}$ is a *regularization function* that measures the complexity of the model. This term is used to penalise models that are too complex leading to over-fitting on the training data at the expense of poor generalisation. Regularization approaches have been shown to be appropriate methodologies for dealing with high-dimensional modeling problems [12]. Parameter, $\lambda \geq 0$ is a *hyperparameter* that provides a trade-off between the two terms and is normally estimated using an outer cross-validation optimization loop, that is $\lambda^* = \arg\min_\lambda \mathcal{F}^*(\theta_\lambda^*)$ where $\mathcal{F}^*$ denotes $\mathcal{F}$ evaluated over the test dataset $\mathcal{S}^*$.

While a wide variety of choices exist for the model structure $f(\Phi, \theta)$, the cost function $\mathcal{F}$ and the regularization function $\mathcal{R}$, in practice choices are restricted to a limited number of options to ensure that the resulting optimisation problems are convex with respect to θ, and hence tractable. In particular, if f is selected to

be a linear-in-the-parameter model $f(\Phi, \theta) := \Phi\theta$, $\mathcal{F}$ is defined as the sum of squared estimation errors

$$\mathcal{F} := (Y - f(\Phi, \theta))^{\mathsf{T}}(Y - f(\Phi, \theta)), \tag{4}$$

and the regularisation term is defined as

$$\mathcal{R}(\theta) := \theta^{\mathsf{T}}\theta, \tag{5}$$

we obtain the classical Ridge Regression problem formulation. Under these conditions (2) has a single global solution given by

$$\theta^*_\lambda = (\Phi^{\mathsf{T}}\Phi + \lambda I)^{-1}\Phi^{\mathsf{T}}Y. \tag{6}$$

The Ridge Regression formulation can be extended to cover nonlinear regression models f without giving up the desirable convexity features of the optimization problem, by employing the so-called *kernel trick* [17] to embed a nonlinear projection of Φ in a Reproducing Kernel Hilbert Space (RKHS) [26] resulting in a linear regression problem. This is achieved by expressing the Ridge Regression solution in dual form

$$\theta^* = \Phi^{\mathsf{T}}(\Phi\Phi^{\mathsf{T}} + \lambda I)^{-1}Y$$

allowing the prediction of new observations to be

$$\begin{aligned} f(\Phi_{new}) &= \Phi_{new}\Phi^{\mathsf{T}}(\Phi\Phi^{\mathsf{T}} + \lambda I)^{-1}Y \\ &= \langle \Phi_{new}, \Phi \rangle \left(\langle \Phi, \Phi \rangle + \lambda I\right)^{-1}Y. \end{aligned}$$

Other choices exist for the regularisation term, for example, LASSO and Elastic Net [30]. However, these do not enjoy the desirable property of having a closed-form solution, making them less attractive when considering large scale problems.

Kernel Ridge Regression [28] is then obtained by replacing the dot products $\langle \cdot, \cdot \rangle$ with an appropriate kernel function $\mathcal{K}$. The resulting nonlinear regression model is

$$f(\Phi_{new}) = \mathcal{K}(\Phi_{new}, \Phi)c^*.$$

with linear parameter vector c defined as

$$c^* = (\mathcal{K}(\Phi, \Phi) + \lambda I)^{-1}Y.$$

III. Supervised Aggregative Feature Extraction

Building on the concepts introduced in the previous section the proposed supervised aggregative feature extraction (SAFE) methodology will now be presented. We begin by considering the ideal case where we have complete knowledge of the time series functions $x_i^{(j)}(t)$. The classical regression model can then be generalised to the functional regression paradigm by defining f as:

$$f(\mathcal{X}_i, \beta) := \sum_{j=1}^{p} \left\langle x_i^{(j)}(t), \beta^{(j)}(t) \right\rangle_{L^2} \tag{7}$$

where $\langle f, g \rangle_{L^2}$ is the L^2 inner product of real functions f and g, defined as $\langle f, g \rangle_{L^2} = \int_{-\infty}^{\infty} f(t)g(t)dt$ and

$$\begin{aligned} \mathcal{X}_i &= \left[x_i^{(1)}(t),\ x_i^{(2)}(t),\ \dots,\ x_i^{(p)}(t) \right], \\ \beta &= \left[\beta^{(1)}(t),\ \beta^{(j)}(t),\ \dots,\ \beta^{(p)}(t) \right]. \end{aligned}$$

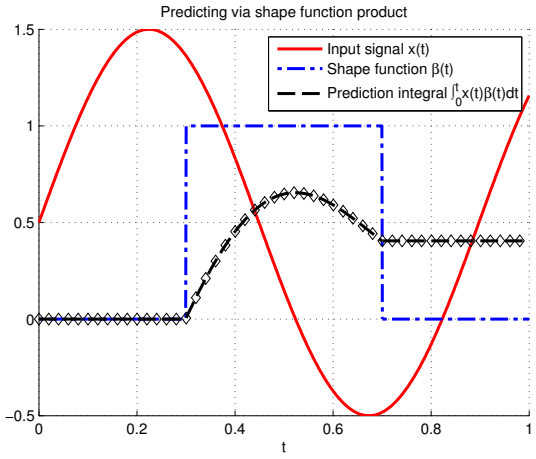


Fig. 1. The input signal (solid line) times the shape function (dash-dotted line) is integrated to obtain the target value. The final value ($t = 1$) of the prediction integral (diamonds) is the prediction output.

In this formulation the regression coefficients generalise to continuous shape functions $\beta^{(j)}(t)$ and the contribution to the prediction of the target output y_i of each time series is obtained as the weighted integration of the time series with the shape function as illustrated in Figure 1. In this setting the modelling task becomes one of estimating the shape functions β. To this end the sum squared error cost function in (4) is generalised to

$$\mathcal{F}(\beta) = \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \int_{-\infty}^{\infty} \beta^{(j)}(t) x_i^{(j)}(t) dt - y_i \right)^2 \tag{8}$$

and the model complexity regularisation function (5) becomes

$$\mathcal{R}(\beta) = \sum_{j=1}^{p} \left\langle \beta^{(j)}, \beta^{(j)} \right\rangle_{L^2},$$

resulting in following functional learning optimization problem for β:

$$\beta^*_\lambda = \arg\min_{\beta}\ \mathcal{L}_\lambda(\beta) = \arg\min_{\beta}\ \mathcal{F}(\beta) + \lambda\mathcal{R}(\beta) \tag{9}$$

A. Practical considerations

In order to make solving (9) tractable a parametrization of the shape functions $\beta^{(j)}(t)$ is adopted. Many possibilities exist including, for example, algebraic and trigonometric polynomials, splines, multilayer perception neural networks and Radial Basis Function (RBF) expansions. Here we adopt an RBF expansion in the form of a linear combination of Gaussian densities to represent $\beta^{(j)}(t)$, that is:

$$\beta^{(j)}(t) = \sum_{k=1}^{\gamma} \alpha_k^{(j)} G(\mu(k), \sigma^2; t) \tag{10}$$

where $\mu(k) = (k-1)/(\gamma - 1)$ and $G(.)$ denotes the Gaussian density function and is defined as

$$G(a, b^2; x) := \frac{1}{\sqrt{2\pi}b} e^{-\frac{(a-x)^2}{2b^2}}. \tag{11}$$

The number of basis functions γ and the bandwidth of each Gaussian density σ^2, which determine the flexibility available to the shape function representation, are assumed to be determined *a priori* in order to yield a linear-in-the-parameter formulation. Several data-driven techniques can be employed to optimally select these parameters (see for example [23] and [3]).

A second practical consideration is that we typically only have a finite number of noisy, irregularly sampled data points for each time series $x_i^{(j)}(t)$, as opposed to their continuous function representations. In order to overcome this issue we introduce a Gaussian Process (GP) approximation to the unobserved time series. Specifically, denoting $\hat{x}_i^{(j)}(t)$ as the estimate of the unobserved $x_i^{(j)}(t)$, we can write

$$\hat{x}_i^{(j)}(t) = \sum_{s=1}^{\mathcal{N}_{i,j}} \mathcal{K}(t, t_{i,s}^{(j)}) c_{i,s}^{(j)}, \tag{12}$$

where $\mathcal{K}$ is a suitably chosen kernel function. Parameter vector $c_{i,\cdot}^{(j)}$ is computed from the available samples of $x_i^{(j)}(t)$ as

$$c_{i,\cdot}^{(j)} = (\mathbf{K} + \xi_j I)^{-1} x_{i,\cdot}^{(j)}, \tag{13}$$

where $k_{w,z} \in \mathbf{K}$ are defined as $k_{w,z} = \mathcal{K}\left(t_{i,w}^{(j)}, t_{i,z}^{(j)}\right)$ and $x_{i,\cdot}^{(j)}$ is the column vector of available time series observations. Considering the radial basis kernel function $\mathcal{K}(t_1, t_2) := \exp\left[-(t_1 - t_2)^2/(2\omega^2)\right]$ and from (11), it follows that

$$\begin{aligned} \mathcal{K}(t_1, t_2) &= \sqrt{2\pi}\omega G(t_1, \omega^2; t_2) \\ \hat{x}_i^{(j)}(t) &= \sqrt{2\pi}\omega_{(j)} \sum_{s=1}^{\mathcal{N}_{i,j}} c_{i,s}^{(j)} G(t_{i,s}^{(j)}, \omega_{(j)}^2; t). \end{aligned} \tag{14}$$

Hyperparameters ξ_j and $\omega_{(j)}^2$ for each time series are determined using standard cross-validation techniques.

B. Closed form solution

Using the aforementioned representations for $\beta^{(j)}(t)$ and $\hat{x}_i^{(j)}(t)$, cost function $\mathcal{F}(\beta)$ (eqt. 8) can be written as

$$\begin{aligned} \hat{\mathcal{F}}(\theta) = \sum_{i=1}^{n} \Bigg(& \sqrt{2\pi} \sum_{j=1}^{p} \omega_{(j)} \sum_{k=1}^{\gamma} \alpha_k^{(j)} \sum_{s=1}^{\mathcal{N}_{i,j}} c_{i,s}^{(j)} \times \\ & \times \int_{-\infty}^{\infty} \left(G(\mu(k), \sigma^2; t) G(t_{i,s}^{(j)}, \omega_{(j)}^2; t) \right) dt - y_i \Bigg)^2 \end{aligned}$$

where $\theta = \begin{bmatrix} \alpha_1^{(1)} & \alpha_2^{(1)} & \cdots & \alpha_k^{(j)} & \cdots & \alpha_\gamma^{(p)} \end{bmatrix}^\mathsf{T}$. A benefit of having employed Gaussian density based representations for $\beta^{(j)}(t)$ and $\hat{x}_i^{(j)}(t)$ is that it allows us to simplify $\hat{\mathcal{F}}(\theta)$ and ultimately achieve a closed form solution for θ. Specifically, using the identity

$$\int_{-\infty}^{\infty} G(a, A; x) G(b, B; x) dx = G(a, A + B; b)$$

for $a, b, x \in \mathbb{R}^p$ and $A, B \in \mathbb{R}^{p \times p}$, which holds for Gaussian density functions [21], we can rewrite $\hat{\mathcal{F}}(\theta)$ as

$$\begin{aligned} \hat{\mathcal{F}}(\theta) = \sum_{i=1}^{n} \Bigg(& \sqrt{2\pi} \sum_{j=1}^{p} \omega_{(j)} \sum_{k=1}^{\gamma} \alpha_k^{(j)} \times \\ & \times \sum_{s=1}^{\mathcal{N}_{i,j}} c_{i,s}^{(j)} G(\mu(k), \sigma^2 + \omega_{(j)}^2; t_{i,s}^{(j)}) - y_i \Bigg)^2. \end{aligned}$$

Now introducing the substitutions

$$\delta_{i,s}^{(j)}(k) = \sqrt{2\pi} c_{i,s}^{(j)} \omega_j G(\mu(k), \sigma^2 + \omega_{(j)}^2; t_{i,s}^{(j)}) \tag{15}$$

$$\overline{\delta}_i^{(j)}(k) = \sum_{s=1}^{\mathcal{N}_{i,j}} \delta_{i,s}^{(j)}(k), \tag{16}$$

we obtain

$$\hat{\mathcal{F}}(\theta) = \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \sum_{k=1}^{\gamma} \alpha_k^{(j)} \overline{\delta}_i^{(j)}(k) - y_i \right)^2. \tag{17}$$

which in turn can be expressed in matrix form as

$$\hat{\mathcal{F}}(\theta) = ||\Phi\theta - Y||^2$$

with regressor matrix Φ defined as

$$\Phi = \begin{bmatrix} \overline{\delta}_1^{(1)}(1) & \dots & \overline{\delta}_1^{(1)}(\gamma) & \overline{\delta}_1^{(2)}(1) & \dots & \overline{\delta}_1^{(p)}(\gamma) \\ \vdots & & \vdots & \vdots & & \vdots \\ \overline{\delta}_i^{(1)}(1) & \dots & \overline{\delta}_i^{(1)}(\gamma) & \overline{\delta}_i^{(2)}(1) & \dots & \overline{\delta}_i^{(p)}(\gamma) \\ \vdots & & \vdots & \vdots & & \vdots \\ \overline{\delta}_n^{(1)}(1) & \dots & \overline{\delta}_n^{(1)}(\gamma) & \overline{\delta}_n^{(2)}(1) & \dots & \overline{\delta}_n^{(p)}(\gamma) \end{bmatrix}.$$

In a similar fashion the regularisation penalty $\mathcal{R}(\beta)$ reduces to

$$\hat{\mathcal{R}}(\theta) = \sum_{j=1}^{p} \left(\sum_{i=1}^{\gamma} \sum_{k=1}^{\gamma} \alpha_i^{(j)} \alpha_k^{(j)} G(\mu(i), 2\sigma^2, \mu(k)) \right).$$

This can be expressed in matrix form as $\hat{\mathcal{R}}(\theta) = \theta^\mathsf{T} \mathbf{D} \theta$ where $\mathbf{D} = diag(D^{(1)}, D^{(2)}, ..., D^{(p)})$ is a block diagonal matrix with block matrices $d_{i,k} \in D^{(j)} \in \mathbb{R}^{\gamma \times \gamma}$ defined as $d_{i,k} = G(\mu(i), 2\sigma^2, \mu(k))$. Since $\hat{\mathcal{F}}$ and $\hat{\mathcal{R}}$ are both quadratic in θ the resulting optimization problem (weighted ridge regression) has an analytical solution

$$\theta_\lambda^* = (\Phi^\mathsf{T}\Phi + \lambda \mathbf{D})^{-1} \Phi^\mathsf{T} Y. \tag{18}$$

In practice, due to the locality of support of Gaussian basis functions, $\mathbf{D}$ is diagonally dominant, hence $\hat{\mathcal{R}}$ can be approximated as (5) with the corresponding ridge regression solution given by (6).

C. Incorporating derivative information

One of the added benefits of the SAFE methodology is that time series derivative information can easily be included in the modelling process. This is facilitated by the availability of functional expressions for the time series (eq. 14). In particular, taking advantage of the

properties of the Gaussian density function [21], the first and second derivative of $\hat{x}_i^{(j)}(t)$, can be computed as

$$\frac{\partial \hat{x}_i^{(j)}(t)}{\partial t} = -\frac{\sqrt{2\pi}}{\omega_{(j)}} \sum_{s=1}^{\mathcal{N}_{i,j}} G(t_{i,s}^{(j)}, \omega_{(j)}^2; t) c_{i,s}^{(j)} (t - t_{i,s}^{(j)}),$$

$$\frac{\partial^2 \hat{x}_i^{(j)}(t)}{\partial^2 t} = \frac{\sqrt{2\pi}}{\omega_{(j)}^3} \sum_{s=1}^{\mathcal{N}_{i,j}} G(t_{i,s}^{(j)}, \omega_{(j)}^2; t) c_{i,s}^{(j)} ((t - t_{i,s}^{(j)})^2 - \omega_{(j)}^2),$$

respectively. These terms can then be introduced as additional features in an expanded Φ (see [21]).

D. Computational complexity

The SAFE methodology has three major computational components: (1) fitting GP models to each time series; (2) estimating the shape function parameters; (3) hyperparameter optimisation. Components (1) and (2) both involve large matrix inversions, namely, equation (13) for the GP models and equation (18) for the shape functions. These have $\mathcal{O}(\mathcal{N}_{i,j}^3)$ and $\mathcal{O}(p^3\gamma^3)$ complexity, respectively. The GP hyperparameters (ξ_j and $\omega_{(j)}^2$) and shape function hyperparameters (γ, σ^2 and λ) can be optimised through outer cross-validation optimisation loops involving multiple repetitions of (1) and (2). Therefore, denoting the number of repetitions of each step needed as q and r, respectively, the overall computational complexity of SAFE is $\mathcal{O}(qnp\mathcal{N}_{i,j}^3 + r(p\gamma)^3)$.

Note, that the values of q and r are determined by the resolution we require when optimising the hyperparameters. In practice this can be quite low. Furthermore, good hyperparameter estimates can often be determined off-line using heuristic techniques or by conducting a pilot study, leaving only the linear parameter estimation steps with an overall complexity of $\mathcal{O}(np\mathcal{N}_{i,j}^3 + p^3\gamma^3)$.

IV. Simulation Results

In this section, the capabilities of SAFE (with and without the time series derivative extension) are demonstrated using three specially constructed synthetic case study datasets. Comparative results are provided for a number of alternative time series feature extraction methodologies as follows:

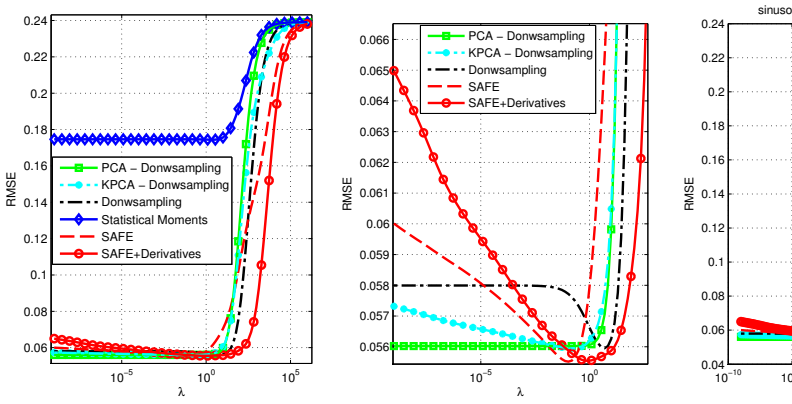
- *Statistical moments*: The first 4 global statistical moments of each time series are used as features (i.e. $k_{\text{MAX}} = 4$ and $\mathcal{M} = 1$ as introduced in Section I)
- *Downsampling*: Time series are partitioned into $\mathcal{M}$ intervals and each interval is represented by its mean value (i.e. $k_{\text{MAX}} = 1$ and $\mathcal{M} = 10$)
- *PCA - Downsampling*: This is an enhancement of downsampling where redundancy in the resulting design matrix is eliminated by replacing the matrix with its r most significant principal components, as determined using Principal Component Analysis (PCA) [12]. For convenience r was fixed as $\mathcal{M}/2$, but can also be chosen as a function of the explained variance. For the problems considered here the choice of $\mathcal{M}/2$ ensures that the PCA summary captures greater than 97% of the variance in the data.
- *KPCA - Downsampling*: This is simply PCA - Downsampling using the kernelized extension of PCA (KPCA) [22]. KPCA produces non-linear transformations of the data and hence can produce more efficient representations if underlying relationships are non-linear. Here, we employ Gaussian kernels.

For consistency with SAFE the design matrices generated by each methodology are used to develop ridge regression based linear models as described in Section II. The Root Mean Squared prediction error (RMSE) of these models, computed on test data and averaged over 500 Monte Carlo simulations, is then used as the performance metric for comparisons.

The three primary synthetic datasets are single time series datasets ($p = 1$), consisting of 150 observations of between 35 and 45 samples of an input time series (uniformly sampled) and corresponding target outputs. Both the input and output samples are subject to Gaussian distributed white noise with expected value 0 and standard deviation 0.1. Multiple time series extensions of two of the datasets are similarly specified. For modelling purposes the data is split into training and test data sets on a 2 to 1 basis (i.e. $n = 100, n^* = 50$). The dimensionality of the datasets in the simulation experiments and also the real industrial use cases presented in Section V are summarized in Table I.

Exp. Name	Type	n	p	$\mathcal{N}_{ij}$	Section
Sinusoid #1	Synthetic	150	1	[35, 45]	IV-A
Sinusoid #2	Synthetic	150	15	[35, 45]	IV-B
Ramp #1	Synthetic	150	1	[35, 45]	IV-B
Ramp #2	Synthetic	150	30	[35, 45]	IV-B
Exponential	Synthetic	150	1	[35, 45]	IV-C
Exp.1	Industrial	1747	2024	57	V
Exp.2	Industrial	1747	2024	30	V

TABLE I
DIMENSIONS OF THE SIMULATED AND INDUSTRIAL DATASETS

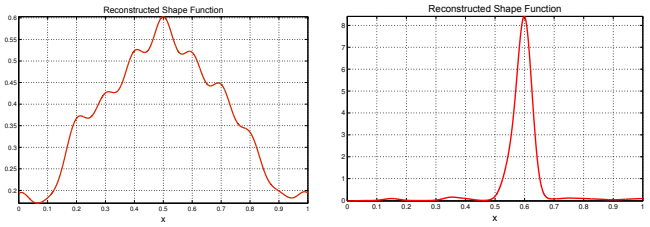


2.a. Original View 2.b. Zoomed View

Fig. 2. Sinusoid dataset results (average over 500 simulations).

A. The sinusoid dataset

This dataset is designed to simulate a scenario where only an unknown part of the input time series deter-



3.a. a=0.3, b=0.7, $\mathcal{U}$(0.01,10) 3.b. a=0.5, b=0.7, $\mathcal{U}$(8,10)

Fig. 3. Shape functions (median models over the Monte Carlo simulations) for the sinusoid dataset with different experimental settings.

mines the target output. The i-th observation of the input time series is defined as:

$$x_i(t) = \sin(\omega_i t + \delta_i) \tag{19}$$

where $\omega_i \sim \mathcal{U}(0.01, 10)$ and $\delta_i \sim \mathcal{U}(0, 2\pi)$, and the output is defined as

$$y_i = \int_a^b x_i(t)dt = \frac{\cos(\omega_i a + \delta_i) - \cos(\omega_i b + \delta_i)}{\omega_i}.$$

Figure 2.a. shows the average RMSE performance of each method with this dataset when $a = 0.3$ and $b = 0.7$. Results are plotted as a function of the regularisation hyperparameter λ so that the sensitivity of performance to this parameter can be observed. As expected, with the exception of the statistical moment-based feature extraction methodology (RMSE=0.18), all methods perform well for this scenario (RMSE<0.06), with the SAFE methodology yields marginally better optimum results on average[1] (Figure 2.b.).

In addition to providing improved features for modelling, SAFE can also enhance model interpretability by highlighting what parts of a time series are most important. This information is revealed through analysis of the shape functions $\beta(t)$ generated. For example, as can be observed in Fig. 3.a., the shape function estimated for this example correctly identifies the central region of the time series as the most relevant for predicting y. As a second example the plot in Fig. 3.b. shows the shape function obtained for the sinusoidal dataset when $a = 0.5$, $b = 0.7$ and $\mathcal{U}(8, 10)$.

A second experiment has been performed with the sinusoid dataset to test the SAFE methodology with multi-input data. The new dataset consists of $p = 15$ time-series $x^{(j)}$ where the i-th observation is defined as in (19), while the output is

$$y_i = \sum_{j=1}^{p} c_j \int_a^b x_i^{(j)}(t)dt,$$

with $c_j \sim \mathcal{B}$ a Bernoulli distribution with success probability equals to 0.35. The RMSE performances at the optimal value of λ (i.e. the value that yields the minimum RMSE on average for each methodology) with this dataset are reported in boxplot form in Fig. 4. The results

[1]At the optimal value of λ the standard deviation of the predictions are below 0.0007 for all the considered methodologies.

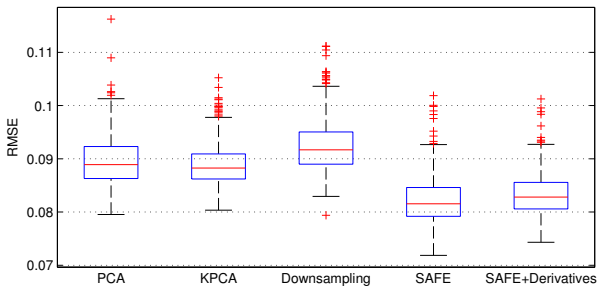


Fig. 4. Multi-dimensional Sinusoid dataset results: RMSE at the optimal value of λ for 500 Monte Carlo simulations

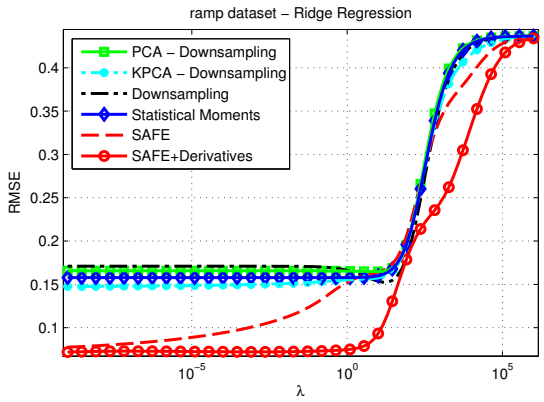


Fig. 5. Ramp dataset results (average over 500 simulations)

show that the SAFE methodologies outperform the other methods on average. In fact, over the 500 Monte Carlo simulations SAFE achieves the minimum RMSE 95.4% of the time (SAFE 59.4% and SAFE+Derivatives 36%, while Downsampling, PCA and KPCA are the best methods 0.4%, 2.4% and 1.8% of the times, respectively.

B. The ramp dataset

The ramp dataset is synthesised to highlight the benefits of including time series derivative information as features in the design matrix. Accordingly, the input time series is generated as

$$x_i(t) = \begin{cases} \zeta_i \sqrt{2t} & t < 0.5 \\ \zeta_i + \phi_i(t - 0.5) & t \geq 0.5 \end{cases} \tag{20}$$

with $\zeta_i \sim \mathcal{U}(0, 1)$, $\phi_i \sim \mathcal{U}(1, 4)$ and the output as $y_i = \phi_i$. Hence, the target output is the slope of $x_i(t)$ in the interval $1 \geq t \geq 0.5$. Figure 5 shows the substantially superior results obtained for this problem using SAFE by virtue of being able to include derivative information as input features in the dataset.

A second experiment has been performed with the ramp dataset to test the SAFE methodology in a multi-input setting. The new dataset consists of $p = 30$ time-series $x^{(j)}$ where the i-th observation is defined as

- $x_i^{(1)}$ as in (20);

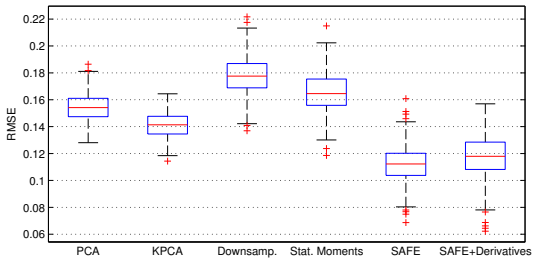


Fig. 6. Multi-dimensional Ramp dataset: RMSE at the optimal value of λ for 500 Monte Carlo simulations

- $x_i^{(j)}$ with $j > 1$ is a 1-D discrete random walk with $x_i^{(j)}(0) \sim \mathcal{U}(0,1)$ step size $\Delta t = 0.01$ and $x_i^{(j)}(\Delta t) \sim N(0, 0.0025)$;

and the target of the model is again the slope of $x_i^{(1)}(t)$ in $[0.5, 1]$ ($y_i = \phi_i$).

The performance of the various feature extraction methodologies with this dataset is compared in Fig. 6, which shows the boxplot of the distribution of RMSE values at the optimal λ for each approach. Although there is some deterioration in performance compared to the singe ramp time-series dataset (the ones reported in Fig. 5), SAFE continues to yield the best performance.

C. The exponential dataset

The purpose of the exponential dataset is to test the performance of the SAFE methodology when the target variable is entirely explained by global features of the input time series. As such the input series is defined as $x_i(t) = a_i e^{-b_i t}$, with $a_i \sim \mathcal{U}(8, 12)$, $b_i \sim \mathcal{U}(0.5, 1.5)$ and the target output to be predicted is given by

$$\begin{aligned} y_i &= \int_0^1 \left(x_i(t) - \int_0^1 x_i(t)dt \right)^2 dt \qquad (21) \\ &= \frac{a_i^2 \left(1 - e^{-2\,b_i}\right)}{2\,b_i} - \frac{a_i^2 \left(e^{-2\,b_i} - 2e^{i\,b} + 1\right)}{b_i^2}. \end{aligned}$$

In this scenario the output, as defined by Equation (21), is the expected value of the second-order sample statistical moment of the observed data (sample variance), hence one would expect that using statistical moment features should yield the best results. However, while the statistical moment features substantially outperform the PCA, KPCA and Downsampling feature extraction methodologies, Figure 7 shows that the SAFE technique again yields the best prediction performances. This somewhat counter-intuitive result arises because of the high variance of the sample second-order central moment estimator at low sample sizes. As illustrated in Figure 8, at the sample sizes defined for this dataset (35 to 45) the estimator is highly imprecise. This dataset therefore shows how the SAFE methodology, albeit relying exclusively on local features, can outperform methods that generate globally defined features even when the target phenomenon is global by its very definition.

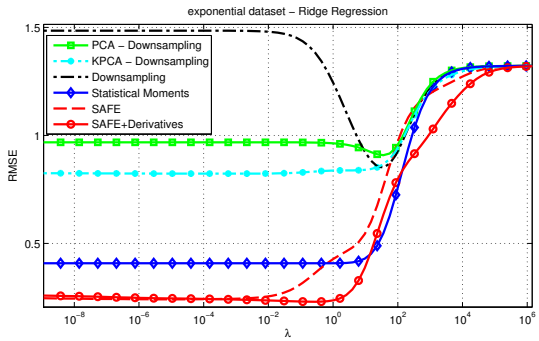


Fig. 7. Exponential dataset results (average over 500 simulations)

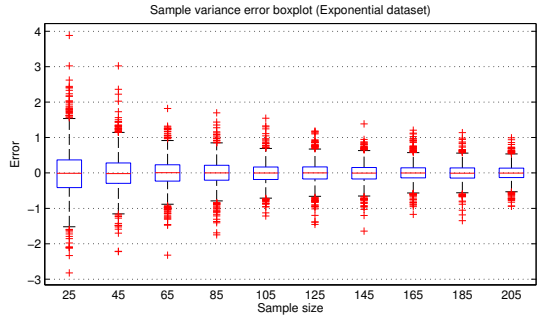


Fig. 8. Exponential dataset: sample variance error boxplots for different sample sizes

D. Sensitivity to sampling rate

It is expected that, thanks to its built-in noise filtering and smoothing capabilities, the proposed SAFE methodology can cope with irregular sampling intervals in an efficient way. While studies regarding the impact of intermittent sampling rate on filtering exist in the literature [24], their effects on feature extraction quality are largely unexplored.

In order to gauge the performance of the proposed algorithm in such situations, the previously described "ramp dataset experiment" was repeated with increasingly variable sampling rates. Specifically, the variance of the sampling time interval $\Delta t = t_{i+1} - t_i$ was defined as e^k for $k = 1, \ldots, 10$. In each case the resulting time vectors were normalized so that $0 \leq t \leq 1$. Hence, the experiments corresponding to low values of k will have almost homogeneous sampling times, while those with high values of k will have very variable sampling times. Figure 9 shows the results obtained with SAFE under these conditions. As anticipated, there is no significant degradation in performance with increasing sampling variability. The worst data points of the most extreme experiments only show a factor of 2.3 increase in RMSE relative to the mean performance (0.16 versus 0.07).

V. Industrial Case Study

As a practical demonstration of SAFE we consider in this section its application to a benchmark soft sensing problem from semiconductor manufacturing, namely

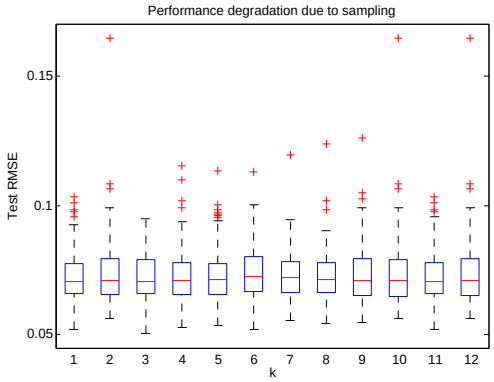


Fig. 9. Ramp dataset: RMSE performance as a function of increasing sample rate variability

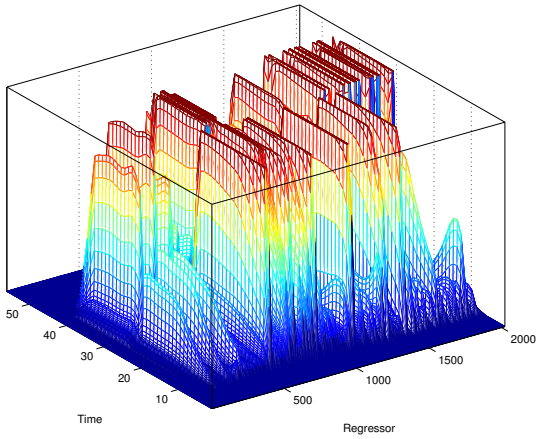


Fig. 10. A typical OES spectrum from a plasma etch process

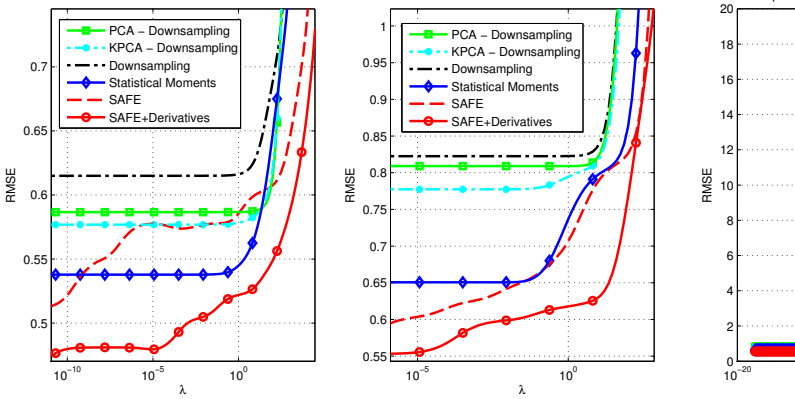
estimating the etch rate of a plasma etch processing tool from Optical Emission Spectrometry (OES) measurements recorded during processing [13], [18].

Plasma etching is ubiquitous in modern semiconductor manufacturing due to its ability to provide precise control of the resolution and directionality of etch. In a typical plasma etching processing tool [11] gases are introduced into a vacuum chamber and ionized to generate a plasma which then interacts both chemically and mechanically with the masked wafer surface to etch away the exposed surface. To achieve the desired precision on critical feature dimensions the key parameter which needs to be controlled is etch rate [18], but this information is not available in real-time as it can only be determined through a costly post processing metrology step. However, using optical emission spectroscopy (OES) monitoring of the plasma, which allows the changes in the plasma chemistry during etching to be observed indirectly, soft sensing solutions can be developed for etch rate prediction.

The benchmark industrial dataset available for this case study consists of OES spectra ($\mathcal{X}$) for a total of $n = 1747$ etch process runs together with associated actual etch rate (y). Each OES spectrum presents us with $p = 2024$ time series corresponding to the evolution of the individual spectrometer channels (spectrum wavelengths) for the duration of the process in question. The spectrum output is available through a set of $\mathcal{N}_{i,j} = 57$ equally spaced samples for each channel. A typical spectrum is plotted in Fig. 10 for a single process observation.

As was the case with the simulated datasets, Monte Carlo simulations (5000 repetitions) are used in evaluating the performance of the various feature extraction methodologies considered, this time using repeated random sub-sampling validation [20] to generate the different instances of the training (60%), validation (20%) and test (20%) datasets.

Two different experiments were conducted. In the first (*Exp.1*) the SAFE methodology was applied with $\gamma = 10$ base Gaussian components selected for the shape functions (equation 10) and the bandwidth σ^2 chosen in order



11.a. *Exp.1* 11.b. *Exp.2*

Fig. 11. Semiconductor manufacturing dataset: RMSE as a function of the regularization parameter λ

to have as close as possible to a uniform distribution when the coefficients $\alpha_k^{(j)} = 1, \forall k = 1, \ldots, \gamma$ (i.e. giving *a priori* equal importance to all time points); the regularization ξ_j and kernel bandwidth $\omega_{(j)}^2$ hyperparameters for each time series were optimised using cross-validation on the validation data sets.

Fig. 11.a. shows the average test data RMSE performance of the Ridge Regression etch rate prediction models obtained with each of the feature extraction methodologies investigated for *Exp.1*; it can be appreciated how SAFE outperforms the other techniques in terms of the minimum RMSE prediction error achieved. In particular, SAFE augmented with OES time series derivatives consistently outperforms all the other methods across the full range of regularization parameter considered, with a relative improvement in the minimum RMSE of 11.7% w.r.t. Statistical Moments, 19% w.r.t. PCA, 17.7% w.r.t KPCA and 22.8% w.r.t. Downsampling.

A second experiment (*Exp.2*) was performed to simulate the scenario of missing measurements or irregularly sampled time series data. This is achieved by randomly retaining 30 out of the 57 available samples for each process run. The experimental settings are the same as

Method	Exp. 1	Exp. 2	Degradation [%]
PCA - Downsampling	0.5866	0.8091	37.9%
KPCA - Downsampling	0.5768	0.7774	34.8%
Statistical Moments	0.5378	0.6507	21.0%
Downsampling	0.6150	0.8226	33.8%
SAFE	0.5132	0.5956	16.1%
SAFE+Derivatives	0.4747	0.5535	16.6%

TABLE II
RMSE DEGRADATION FROM *Exp.1* TO *Exp.2*

in *Exp.1* with $\gamma = 10$, bandwidth σ^2 chosen to provide the same *a priori* importance to all time points and ξ_j and $\omega^2_{(j)}$ optimised through cross-validation.

The average test data RMSE performance of each model is reported in Fig. 11.b.. Again it can be seen that SAFE and SAFE augmented with OES time series derivative information outperform the other feature extraction approaches for all values of λ considered.

Comparing Fig.11 (*Exp.2*) with Fig.10 (*Exp.1*) allows the impact of employing irregular/reduced sampling to be assessed. Table II provides a numerical comparison of the two experiments and quantifies the degradation in performance observed with *Exp.2* relative to *Exp.1*. As expected, the performance of all the approaches degrades in *Exp.2*, but the SAFE implementations are the least impacted, with a 16% increase in RMSE relative to *Exp.1*, compared to 21% with statistical moments and greater than 33% with the other approaches. The results thus confirm that SAFE is more robust to sampling variability than the other methods evaluated.

VI. CONCLUSIONS

With many industries investing heavily in advanced process monitoring technologies and infrastructure to support collection, integration and archiving of process data from heterogeneous sources, the future is *big data*. This brings both opportunities and challenges; opportunities such as eliminating costly metrology using soft sensors and optimising maintenance scheduling using predictive maintenance models; and challenges such as dealing with the data deluge and making the best use of the available data.

In this paper, a novel SAFE methodology has been defined and presented that addresses these challenges for modelling problems where a scalar output to be predicted is a function of one or more time series data streams. The SAFE methodology has two main characteristics; (1) it employs a holistic method for generating features in a supervised fashion that are optimally orientated towards prediction; and (2), it is able to work with multiple heterogeneous time series signals where each one can have a different sampling rate, be non-uniformly sampled and/or be of different length. This makes its a powerful tool for industrial informatics as many industrial datasets are characterised by the fusion of datastreams with different sampling and duration characteristics. Often significant pre-processing effort is needed to align these disparate date streams. The SAFE methodology alleviates this burden, and thus is a promising tool in the increasingly big data world, where automated approaches are becoming a key requirement. Furthermore, an important consideration when developing risk mitigation strategies for industrial processes is effective process monitoring. SAFE contributes to this goal by providing enhanced models for soft sensing and predictive maintenance.

The proposed methodology derives from a functional learning setting in which the time series input space is reconstructed by means of Gaussian process inference, and the unknown shape function is parametrized as a weighted sum of Gaussian functions. This combination allows for a number of interesting properties, including closed form solution (and hence efficient numerical computation procedures), enhanced interpretability through shape function analysis, easy incorporation of time series derivative information, and the possibility of using the extracted information as input data to other machine learning methodologies.

The capabilities of the SAFE methodology with respect to competing time series feature extraction methodologies have been demonstrated by means of simulated examples and further validated using a practical semiconductor manufacturing soft sensing problem. The prediction optimised features extracted by SAFE come at a price as simpler approaches to feature extraction require less computational effort, however the results presented show that SAFE is able to consistently outperform its competitors over a range of input-output relationship and data conditions, including situations where the target output is determined by global features of the input time series.

While SAFE was originally motivated by modelling problems in batch industrial processes (where the input data is the time evolution of sensor readings during the batch run and the output is a scalar indicator of the final product quality) [9], the methodology is applicable to any time series-intensive learning environment, for example, evoked potential studies in neuroscience [4], dynamic biological process modelling in genetics [1] and financial data analysis in economics [2]. It should also be noted that SAFE was conceived and developed for supervised learning problems; the potential for extending the methodology to unsupervised or semi-supervised problems, and any benefits this might bring, have not been investigated to date.

REFERENCES

[1] Z. Bar-Joseph, A. Gitter, and I. Simon. Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, 13(8):552–564, 2012.

[2] N. Beck and J.N. Katz. Modeling dynamics in time-series-cross-section political economy data. *Annual Review of Political Science*, 14:331–352, 2011.

[3] Z.I. Botev, J.F. Grotowski, and D.P. Kroese. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.

[4] D.P. Burke, S.P. Kelly, P. de Chazal, R.B. Reilly, and C. Finucane. A parametric feature extraction and classification strategy for brain-computer interfacing. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 13(1):12–17, March 2005.

[5] D. Chen, P. Hall, and Hans-Georg H.-G. Müller. Single and multiple index functional regression models with nonparametric link. *The Annals of Statistics*, 39(3):1720–1747, 2011.

[6] H. Chen, R.H.L. Chiang, and V.C. Storey. Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 36(4):1165–1188, 2012.

[7] S. Ding, S. Yin, K. Peng, H. Hao, and B. Shen. A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill. *Industrial Informatics, IEEE Transactions on*, 9(4):2239–2247, 2013.

[8] X. Ding, Y. Tian, and Y. Yu. A real-time big data gathering algorithm based on indoor wireless sensor networks for risk analysis of industrial operations. *Industrial Informatics, IEEE Transactions on*, PP(99):1–1, 2015.

[9] P. Facco, F. Doplicher, F. Bezzo, and M. Barolo. Moving average pls soft sensor for online product quality estimation in an industrial batch polymerization process. *Journal of Process Control*, 19:520–529, 2009.

[10] F. Ferraty and P. Vieu. Additive prediction and boosting for functional data. *Computational Statistics & Data Analysis*, 53(4):1400–1413, 2009.

[11] B. Flynn and S. McLoone. Max separation clustering for feature extraction from optical emission spectroscopy data. *Semiconductor Manufacturing, IEEE Transactions on*, 24(4):480–488, 2011.

[12] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning. Data Mining, Inference and Prediction*. Springer, 2009.

[13] S. Hong and G. May. Neural-network-based sensor fusion of optical emission and mass spectroscopy data for real-time fault detection in reactive ion etching. *Industrial Electronics, IEEE Transactions on*, 52(4):1063–1072, 2005.

[14] G. James and B. Silverman. Functional adaptive model estimation. *Jour. of the American Statistical Association*, 100(470):565–576, 2005.

[15] M. Jian, F. Chu, F. Wang, and W. Wang. On-line batch process monitoring using batch dynamic kernel principal component analysis. *Chemometrics Intelligent Lab. Systems*, 101:110–122, 2010.

[16] S. Kaisler, F. Armour, J.A. Espinosa, and W. Money. Big data: Issues and challenges moving forward. In *System Sciences (HICSS), 2013 46th International Conference on*, pages 995–1004. IEEE, 2013.

[17] N. Kwak. Nonlinear projection trick in kernel methods: An alternative to the kernel trick. *Neural Networks and Learning Systems, IEEE Transactions on*, 24(12):2113–2119, Dec 2013.

[18] S. Lynn, J. Ringwood, and N. MacGearailt. Global and local virtual metrology models for a plasma etch process. *Semiconductor Manufacturing, IEEE Transactions on*, 25(1):94–103, 2012.

[19] K. Ohadi, R. Legge, and H. Budman. Development of a soft-sensor based on multi-wavelength fluorescence spectroscopy and a dynamic metabolic model for monitoring mammalian cell cultures. *Biotechnology and bioengineering*, 112(1):197–208, 2015.

[20] R.R. Picard and R.D. Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, 1984.

[21] A. Schirru, G.A. Susto, S. Pampuri, and S. McLoone. Learning from time series: Supervised aggregative feature extraction. In *51st IEEE Conf. on Decision and Control*, pages 5254–5259, 2012.

[22] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[23] S.J. Sheather and M.C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 683–690, 1991.

[24] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. *Automatic Control, IEEE Transactions on*, 49(9):1453–1464, 2004.

[25] F. Souza and R. Araujo. Online mixture of univariate linear regression models for adaptive soft sensors. *Industrial Informatics, IEEE Transactions on*, 10(2):937–945, May 2014.

[26] B.K. Sriperumbudur, K. Fukumizu, and GRG Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *The Journal of Machine Learning Research*, 12:2389–2410, 2011.

[27] G.A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi. Machine learning for predictive maintenance: a multiple classifiers approach. *IEEE Trans. Industrial Informatics*, 11:812–820, 2015.

[28] V. Vovk. Kernel ridge regression. In *Empirical Inference*, pages 105–116. Springer, 2013.

[29] D. Wang, J. Liu, and R. Srinivasan. Data-driven soft sensor approach for quality prediction in a refining process. *Industrial Informatics, IEEE Transactions on*, 6(1):11–17, 2010.

[30] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Gian Antonio Susto (S'11) received the M.S. degree (cum laude) in control systems engineering and the Ph.D. in Information Engineering from the University of Padova, Padova, Italy, in 2009 and 2013. He is currently a post-doctoral associate at University of Padova and Chief Data Scientist at Statwolf Ltd. He has been a visiting student at the University of California, San Diego (2008-09), and at National University of Ireland, Maynooth (2012) and an Intern Researcher at Infineon Technologies Austria AG, Villach, Austria (2011). He is a recipient of IEEE-CASE Best Student Conference Paper (2011), IEEE/SEMI-ASMC Best Student Paper (2012) and IEEE-MSC Best Student Paper (2012) awards. His research interests include manufacturing data analytics, machine learning, gesture recognition and partial differential equations control.

Andrea Schirru received the M.S. degree (cum laude) in computer science and the Ph.D. in Information Engineering from the University of Pavia, Pavia, Italy, in 2008 and 2011. He is currently a post-doctoral associate at National University of Ireland, Maynooth and Chief Technology Officer at Statwolf Ltd. He has been an Intern Researcher at Infineon Technologies Austria AG, Villach, Austria (2010-2011), and during his studies he has been awarded with the ICINCO Best Student Paper Award (2011) and the Best Student Presentation at the Intel European Research and Innovation Conference (2010).

Simone Pampuri received the M.S. degree in computer science and the Ph.D. in Information Engineering from the University of Pavia, Pavia, Italy, in 2009 and 2012. He is currently a post-doctoral associate at National University of Ireland, Maynooth and Chief Executive Officer at Statwolf Ltd. He has been an Intern Researcher at Infineon Technologies Austria AG, Villach, Austria (2011). His research interests include manufacturing data analytics, machine learning and sentiment analysis.

Seán McLoone (S'94, M'96, SM'02) received a Master of Engineering degree with Distinction and a Ph.D. degree in Control Engineering from Queens University Belfast (QUB), Belfast, Ireland, in 1992 and 1996, respectively. He is currently Professor and Director of the Energy Power and Intelligent Control (EPIC) Research Cluster at QUB. His research interests lie in the general area to data based modelling and analysis of dynamical systems. This encompasses techniques ranging from classical system identification, fault diagnosis, and statistical process control to modern computational intelligence based adaptive learning algorithms and optimization techniques. He is a Past Chairman of the UK and Republic of Ireland (UKRI) Section of the IEEE.